

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

In [2]: df = pd.read_csv('Telco-Customer-Churn.csv')
df

Out[2]:
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	No	No	No	Month-to-month
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	No	No	No	One year
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	No	No	No	Month-to-month
3	7795-CFCOW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	Yes	No	No	One year
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	No	No	No	Month-to-month
...
7030	6840-REBVB	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	...	Yes	Yes	Yes	Yes	One year
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	...	Yes	No	Yes	Yes	One year
7040	4301-JJAZL	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	...	No	No	No	No	Month-to-month
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No	...	No	No	No	No	Month-to-month
7042	3186-AJKEK	Male	0	No	No	66	Yes	No	Fiber optic	Yes	...	Yes	Yes	Yes	Yes	Two year

7043 rows x 21 columns

```
In [3]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype
---  --
 0   customerID            7043 non-null   object
 1   gender                7043 non-null   object
 2   SeniorCitizen         7043 non-null   int64
 3   Partner               7043 non-null   object
 4   Dependents            7043 non-null   object
 5   tenure                7043 non-null   int64
 6   PhoneService          7043 non-null   object
 7   MultipleLines         7043 non-null   object
 8   InternetService       7043 non-null   object
 9   OnlineSecurity        7043 non-null   object
10   OnlineBackup          7043 non-null   object
11   DeviceProtection     7043 non-null   object
12   TechSupport          7043 non-null   object
13   StreamingTV          7043 non-null   object
14   StreamingMovies       7043 non-null   object
15   Contract              7043 non-null   object
16   PaperlessBilling      7043 non-null   object
17   PaymentMethod         7043 non-null   object
18   MonthlyCharges        7043 non-null   float64
19   TotalCharges          7043 non-null   object
20   Churn                 7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB

replacing blanks with 0 and tenure is 0 and no total charges are recorded
```

```
In [4]: df['TotalCharges'] = df['TotalCharges'].replace(' ', '0')
df['TotalCharges'] = df['TotalCharges'].astype(float)

In [5]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype
---  --
 0   customerID            7043 non-null   object
 1   gender                7043 non-null   object
 2   SeniorCitizen         7043 non-null   int64
 3   Partner               7043 non-null   object
 4   Dependents            7043 non-null   object
 5   tenure                7043 non-null   int64
 6   PhoneService          7043 non-null   object
 7   MultipleLines         7043 non-null   object
 8   InternetService       7043 non-null   object
 9   OnlineSecurity        7043 non-null   object
10   OnlineBackup          7043 non-null   object
11   DeviceProtection     7043 non-null   object
12   TechSupport          7043 non-null   object
13   StreamingTV          7043 non-null   object
14   StreamingMovies       7043 non-null   object
15   Contract              7043 non-null   object
16   PaperlessBilling      7043 non-null   object
17   PaymentMethod         7043 non-null   object
18   MonthlyCharges        7043 non-null   float64
19   TotalCharges          7043 non-null   float64
20   Churn                 7043 non-null   object
dtypes: float64(2), int64(2), object(17)
memory usage: 1.1+ MB

In [8]: df.isnull().sum().sum()

Out[8]: 0

In [9]: df.describe()

Out[9]:
```

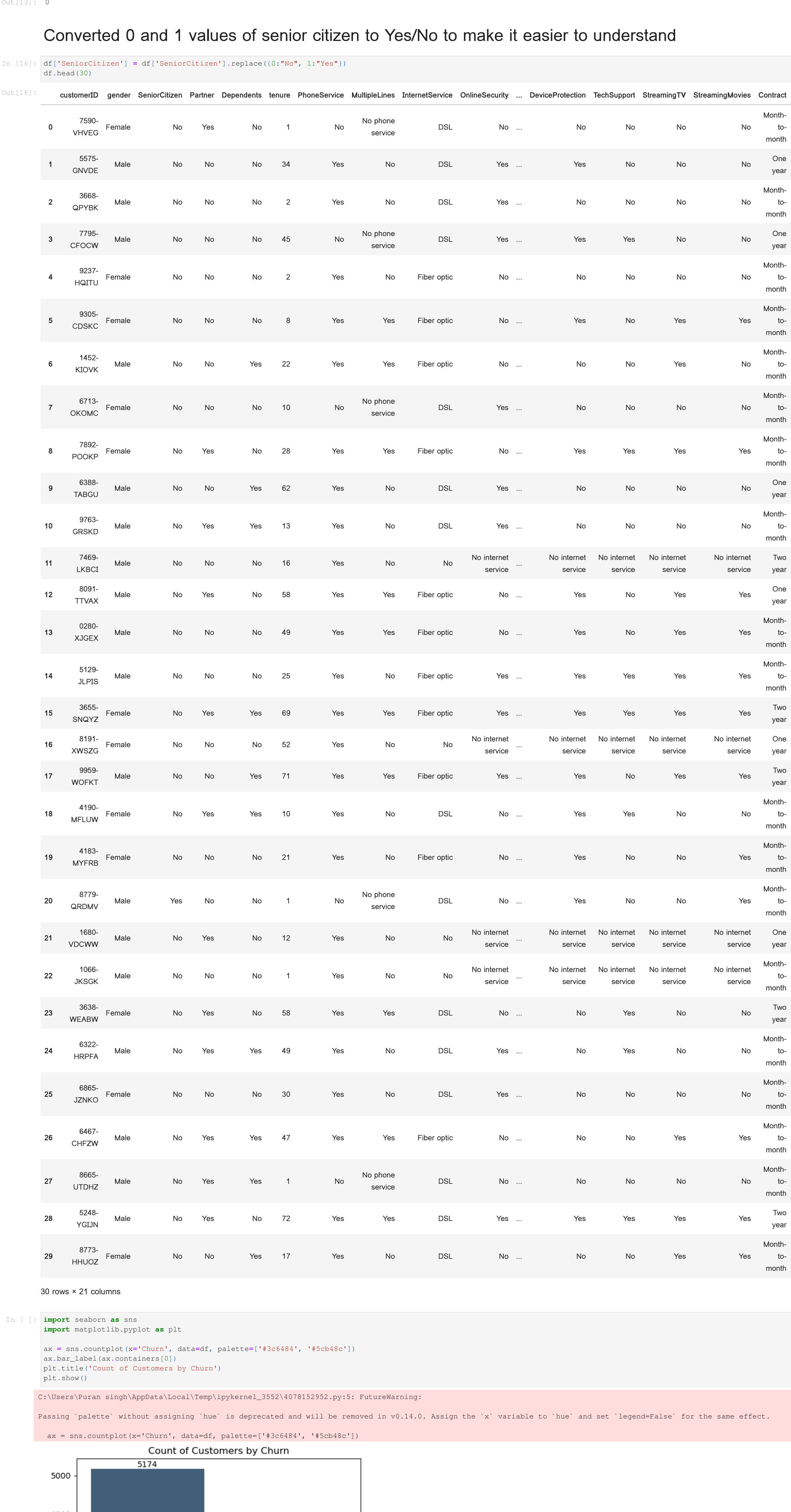
	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
count	7043.000000	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692	2279.734304
std	0.368612	24.559481	30.090047	2266.794470
min	0.000000	0.000000	16.250000	0.000000
25%	0.000000	9.000000	35.500000	396.550000
50%	0.000000	29.000000	70.350000	1394.550000
75%	0.000000	55.000000	89.850000	3786.600000
max	1.000000	72.000000	118.750000	8684.800000

```
In [13]: df['customerID'].duplicated().sum()

Out[13]: 0
```

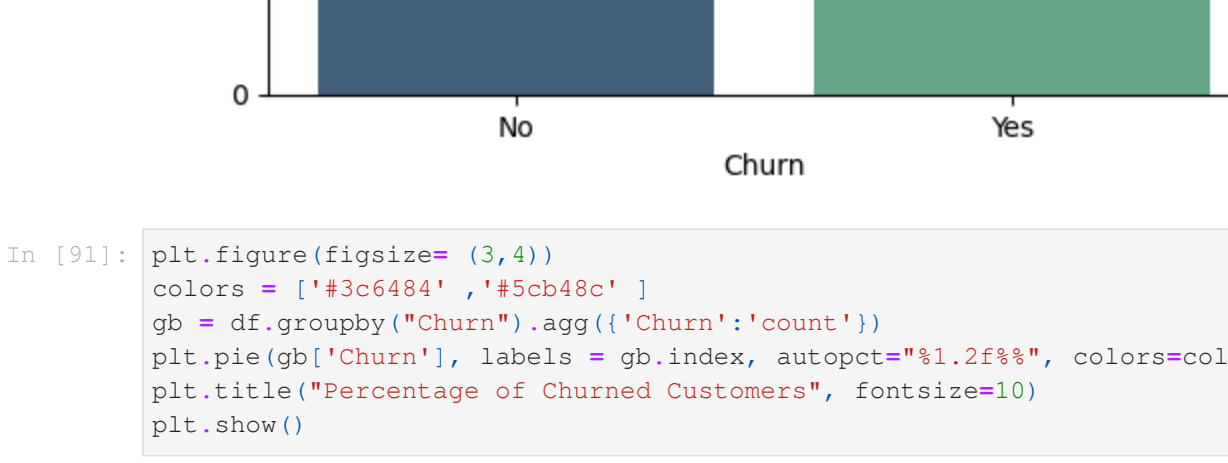
Converted 0 and 1 values of senior citizen to Yes/No to make it easier to understand

```
In [16]: df['SeniorCitizen'] = df['SeniorCitizen'].replace({0:"No", 1:"Yes"})
df.head(30)
```



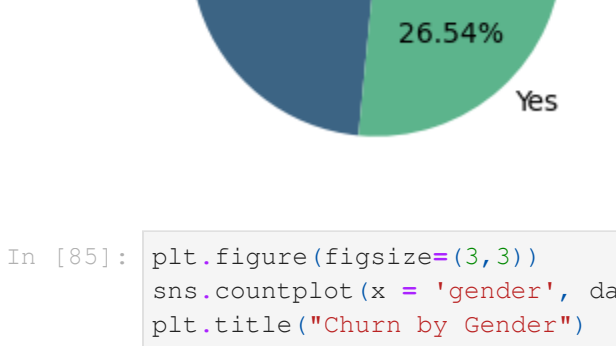
```
In [ ]: import seaborn as sns
import matplotlib.pyplot as plt
ax = sns.countplot(x='Churn', data=df, palette='#3c6484', '#5cb48c')
ax.bar_label(ax.containers[0])
plt.title('Count of Customers by Churn')
plt.show()
```

C:\Users\Puran singh\AppData\Local\Temp\ipykernel_3552\4078152952.py:5: FutureWarning:
Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign the 'x' variable to 'hue' and set 'legend=False' for the same effect.
ax = sns.countplot(x='Churn', data=df, palette='#3c6484', '#5cb48c')

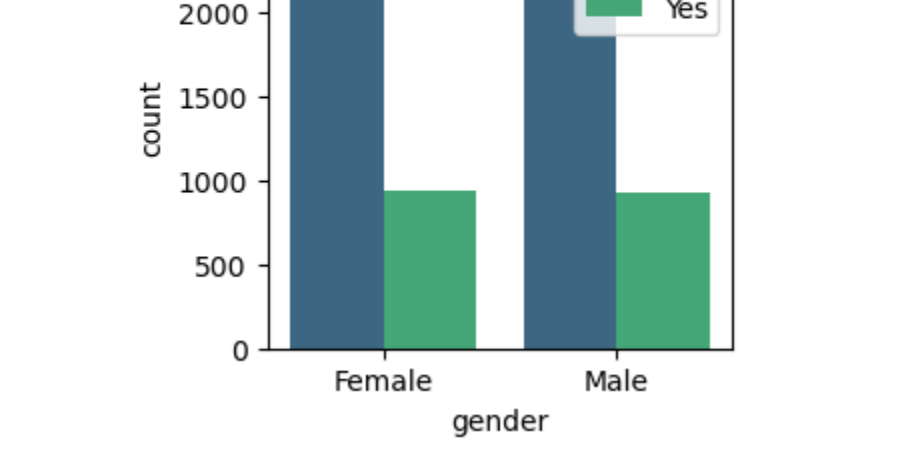


```
In [31]: plt.figure(figsize=(3,4))
colors = ['#3c6484', '#5cb48c']
gb = df.groupby('Churn').agg(['Churn','count'])
plt.gcf().set('Churn').label = gb.index, autogpt='1.21%', colors=colors)
plt.title('Percentage of Churned Customers', fontsize=10)
plt.show()
```

Percentage of Churned Customers

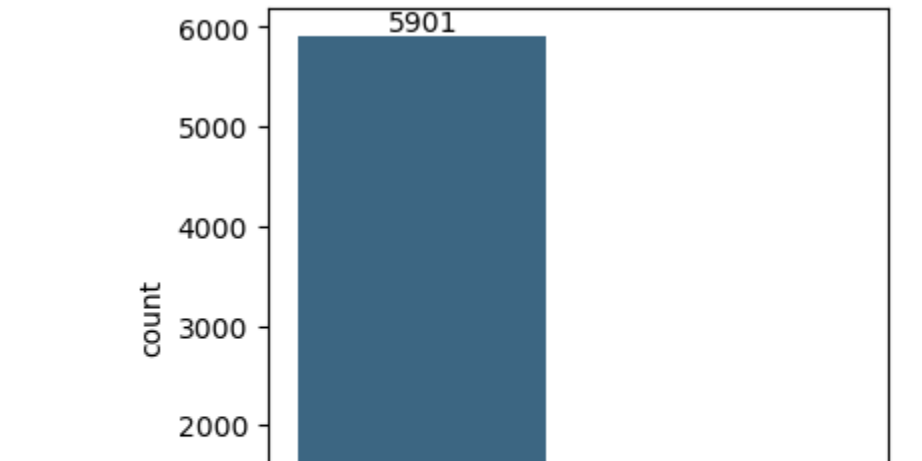


```
In [83]: plt.figure(figsize=(3,3))
sns.countplot(x = 'gender', data=df, hue='Churn', palette='viridis')
plt.title('Churn by Gender')
plt.show()
```



```
In [84]: plt.figure(figsize=(4,4))
ax = sns.countplot(x = 'SeniorCitizen', data=df, palette='viridis')
ax.bar_label(ax.containers[0])
plt.title('Count of customers by Senior Citizen')
plt.show()
```

C:\Users\Puran singh\AppData\Local\Temp\ipykernel_3552\1149669423.py:2: FutureWarning:
Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign the 'x' variable to 'hue' and set 'legend=False' for the same effect.
ax = sns.countplot(x = 'SeniorCitizen', data=df, palette='viridis')



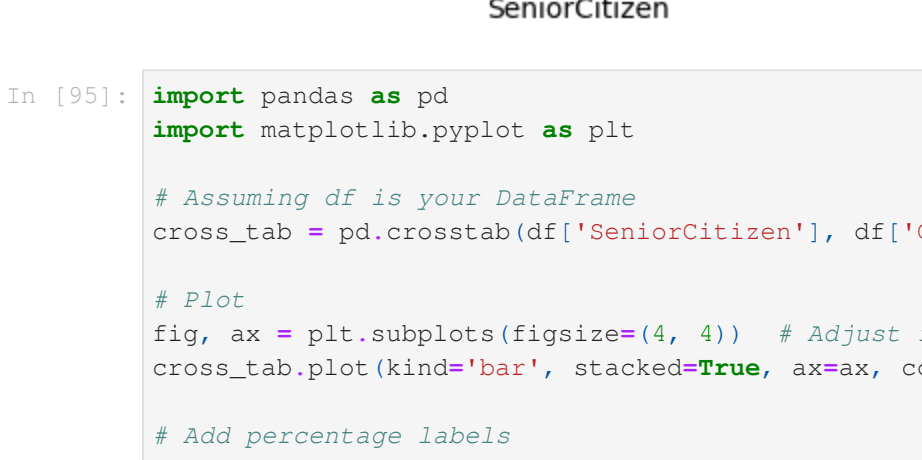
```
In [95]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming df is your DataFrame
cross_tab = pd.crosstab(df['SeniorCitizen'], df['Churn'], normalize='index') * 100

# Plot
fig, ax = plt.subplots(figsize=(4, 4)) # Adjust figure size
cross_tab.plot(kind='bar', stacked=True, ax=ax, color=['#3c6484', '#5cb48c'])

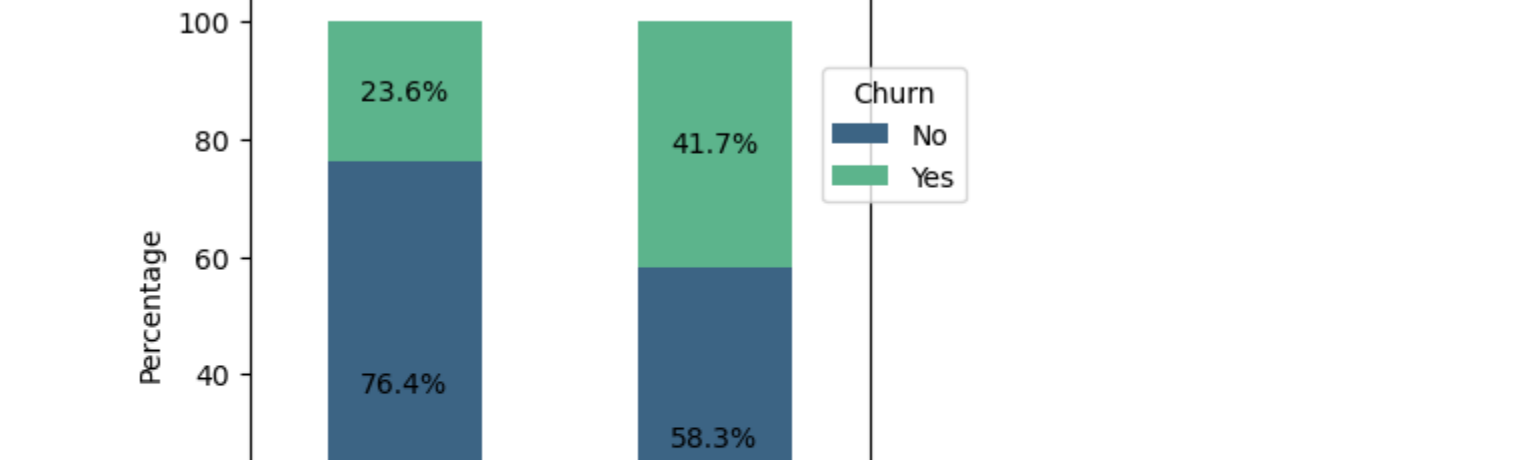
# Add percentage labels
for p in ax.patches:
    width, height = p.get_width(), p.get_height()
    x = p.get_x() + width / 2, y = height / 2, f'{height:.1f}%'
    ax.text(x, y + height / 2, f'{height:.1f}%',
            ha='center', va='center', fontsize=10, color='black')

plt.title('Churn by Senior Citizen')
plt.xlabel('Senior Citizen')
plt.ylabel('Percentage')
plt.legend(title='Churn', bbox_to_anchor=(0.9, 0.9))
plt.show()
```



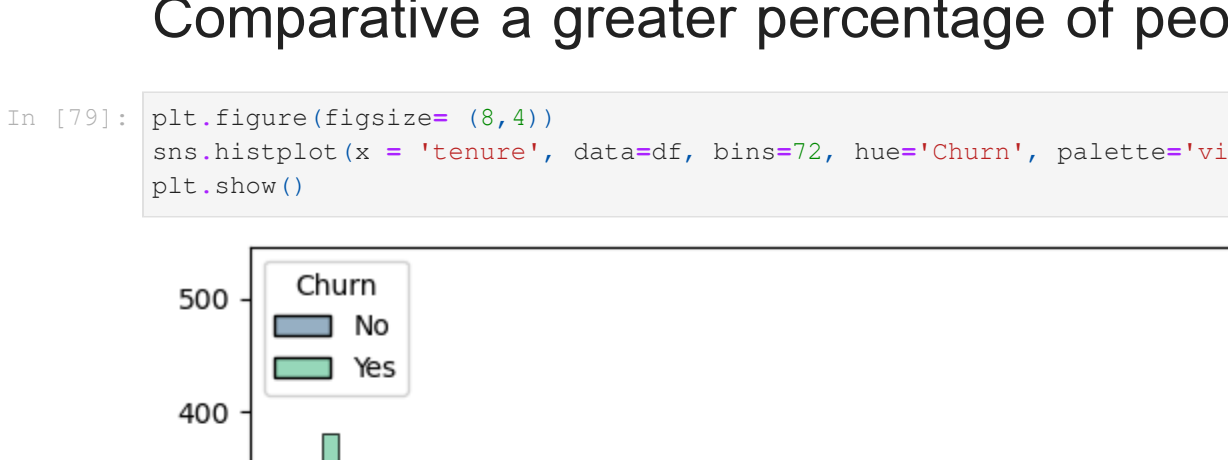
Comparative a greater percentage of people in senior citizen category have churned

```
In [79]: plt.figure(figsize=(6,4))
sns.histplot(x = 'tenure', data=df, bins=72, hue='Churn', palette='viridis')
plt.show()
```



people who have used our services for a long time have stayed and people who have used our services for #1 or #2 months have churned

```
In [78]: plt.figure(figsize=(4,4))
ax = sns.countplot(x = 'Contract', data=df, hue='Churn', palette='viridis')
ax.bar_label(ax.containers[0])
plt.title('Count of customers by Contract')
plt.show()
```



People who have month to month contract are likely are churn then from those who have one or two years of contract

```
Out [57]: df.columns.values

In [57]: array(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
              'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
              'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
              'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
              'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',
              'TotalCharges', 'Churn'], dtype=object)
```

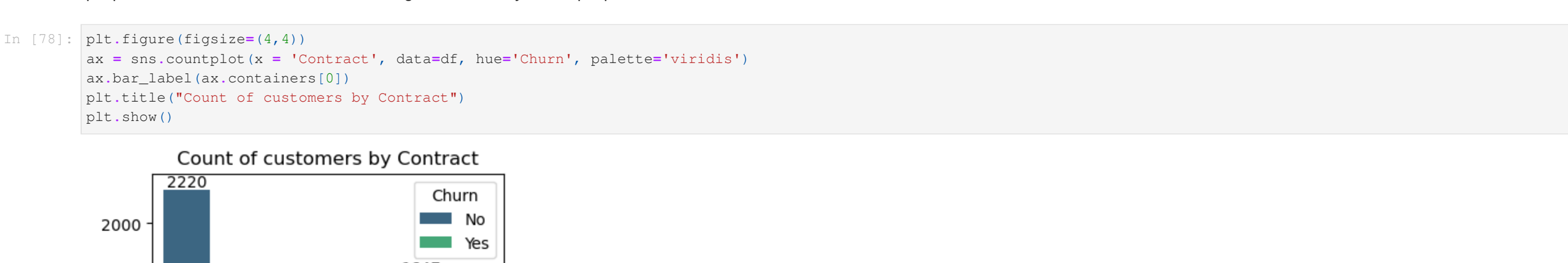
```
In [77]: import matplotlib.pyplot as plt
import seaborn as sns

# Define columns to plot
columns = ('PhoneService', 'MultipleLines', 'InternetService',
          'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
          'TechSupport', 'StreamingTV', 'StreamingMovies')

# Create subplots (3x3 grid)
fig, axes = plt.subplots(3, 3, figsize=(15, 12)) # 3 rows, 3 cols

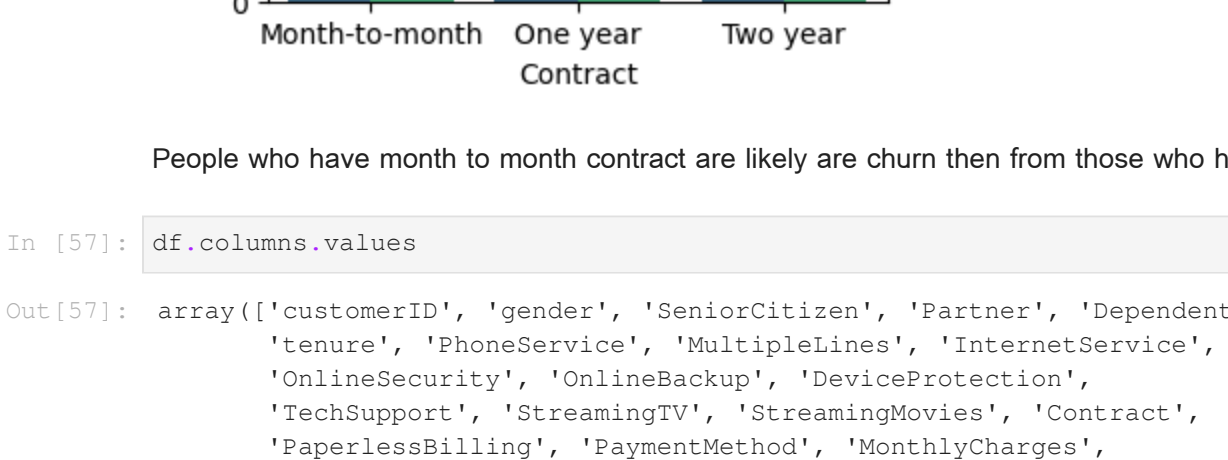
# Flatten axes for easy iteration
axes = axes.flatten()

# Loop through columns and create count plots
for i, col in enumerate(columns):
    sns.countplot(x=df[col], hue='Churn', ax=axes[i], palette='viridis')
    axes[i].set_title(f'Churn by {col}')
    axes[i].set_xlabel('')
    axes[i].set_ylabel('')
    axes[i].set_ylabel('Count')
    axes[i].legend(title='Churn', loc='upper right')
```



The visualizations show the relationship between customer churn and various telecom services. Customers without security, backup, or support services tend to churn more. Fiber optic internet users also exhibit higher churn rates. Streaming services and multiple lines do not significantly impact churn, while phone service users show lower churn rates

```
In [100]: plt.figure(figsize=(6,4))
ax = sns.countplot(x = 'PaymentMethod', data=df, hue='Churn', palette='viridis')
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.title('Churned customers by Payment Method')
plt.xticks(rotation=45)
plt.show()
```



Customer is likely to Churn when he is using electronic check as a payment method